Screenplay Summarization Using Latent Narrative Structure

Screenplay Summarization Using Latent Narrative Structure

Pinelopi Papalampidi[1] **Frank Keller**[1] **Lea Frermann**[2] **Mirella Lapata**[1]
[1]Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh
[2]School of Computing and Information Systems
University of Melbourne
p.papalampidi@sms.ed.ac.uk, keller@inf.ed.ac.uk,
lea.frermann@unimelb.edu.au, mlap@inf.ed.ac.uk

Abstract

Most general-purpose extractive summarization models are trained on news articles, which are short and present all important information upfront. As a result, such models are biased by position and often perform a smart selection of sentences from the beginning of the document. When summarizing long narratives, which have complex structure and present information piecemeal, simple position heuristics are not sufficient. In this paper, we propose to explicitly incorporate the underlying structure of narratives into general unsupervised and supervised extractive summarization models. We formalize *narrative structure* in terms of key narrative events (turning points) and treat it as latent in order to summarize screenplays (i.e., extract an optimal sequence of scenes). Experimental results on the CSI corpus of TV screenplays, which we augment with scene-level summarization labels, show that latent turning points correlate with important aspects of a CSI episode and improve summarization performance over general extractive algorithms, leading to more complete and diverse summaries.

1 Introduction

Automatic summarization has enjoyed renewed interest in recent years thanks to the popularity of modern neural network-based approaches (Cheng and Lapata, 2016; Nallapati et al., 2016, 2017; Zheng and Lapata, 2019) and the availability of large-scale datasets containing hundreds of thousands of document–summary pairs (Sandhaus, 2008; Hermann et al., 2015; Grusky et al., 2018; Narayan et al., 2018; Fabbri et al., 2019; Liu and Lapata, 2019). Most efforts to date have concentrated on the summarization of news articles which tend to be relatively short and formulaic following an "inverted pyramid" structure which places the most essential, novel and interesting elements of a story in the beginning and supporting material and secondary details afterwards. The rigid structure of news articles is expedient since important passages can be identified in predictable locations (e.g., by performing a "smart selection" of sentences from the beginning of the document) and the structure itself can be explicitly taken into account in model design (e.g., by encoding the relative and absolute position of each sentence).

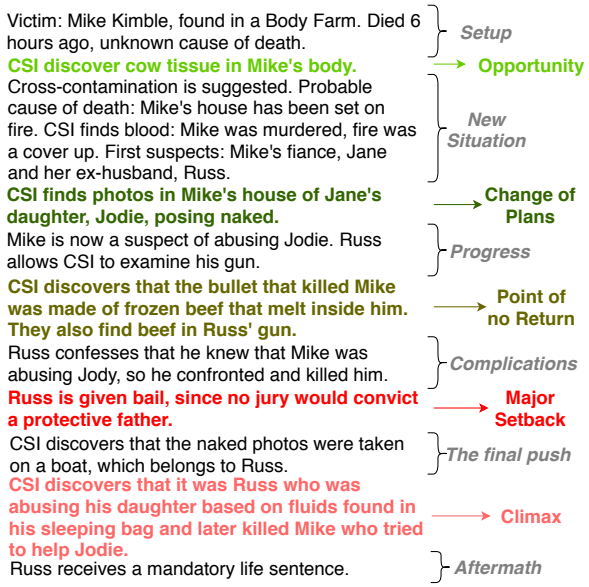

Figure 1: Example of narrative structure for episode "Burden of Proof" from TV series Crime Scene Investigation (CSI); turning points are highlighted in color.

In this paper we are interested in summarizing longer narratives, i.e., screenplays, whose form and structure is far removed from newspaper articles. Screenplays are typically between 110 and 120 pages long (20k words), their content is broken down into scenes, which contain mostly dialogue (lines the actors speak) as well as descriptions explaining what the camera sees. Moreover, screenplays are characterized by an underlying *narrative structure,* a sequence of events by which

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1920–1933
July 5 - 10, 2020.

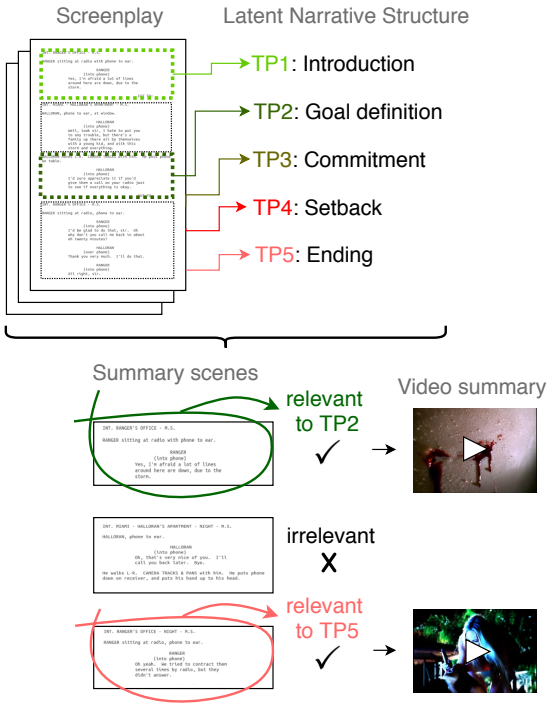

Figure 2: We first identify scenes that act as turning points (i.e., key events that segment the story into sections). We next create a summary by selecting informative scenes, i.e.,semantically related to turning points.

a story is defined (Cutting, 2016), and by the story's characters and their roles (Propp, 1968). Contrary to news articles, the gist of the story in a screenplay is not disclosed at the start, information is often revealed piecemeal; characters evolve and their actions might seem more or less important over the course of the narrative. From a modeling perspective, obtaining training data is particularly problematic: even if one could assemble screenplays and corresponding summaries (e.g., by mining IMDb or Wikipedia), the size of such a corpus would be at best in the range of a few hundred examples not hundreds of thousands. Also note that genre differences might render transfer learning (Pan and Yang, 2010) difficult, e.g., a model trained on movie screenplays might not generalize to sitcoms or soap operas.

Given the above challenges, we introduce a number of assumptions to make the task feasible. Firstly, our goal is to produce *informative* summaries, which serve as a surrogate to reading the full script or watching the entire film. Secondly, we follow Gorinski and Lapata (2015) in conceptualizing screenplay summarization as the task of identifying a sequence of informative scenes. Thirdly, we focus on summarizing television programs such as *CSI: Crime Scene Investigation* (Frermann et al., 2018) which revolves around a team of forensic investigators solving criminal cases. Such programs have a complex but well-defined structure: they open with a crime, the crime scene is examined, the victim is identified, suspects are introduced, forensic clues are gathered, suspects are investigated, and finally the case is solved.

In this work, we adapt general-purpose extractive summarization algorithms (Nallapati et al., 2017; Zheng and Lapata, 2019) to identify informative scenes in screenplays and instill in them knowledge about narrative film structure (Hauge, 2017; Cutting, 2016; Freytag, 1896). Specifically, we adopt a scheme commonly used by screenwriters as a practical guide for producing successful screenplays. According to this scheme, well-structured stories consist of six basic stages which are defined by five *turning points* (TPs), i.e., events which change the direction of the narrative, and determine the story's progression and basic thematic units. In Figure 1, TPs are highlighted for a CSI episode. Although the link between turning points and summarization has not been previously made, earlier work has emphasized the importance of narrative structure for summarizing books (Mihalcea and Ceylan, 2007) and social media content (Kim and Monroy-Hernández, 2015). More recently, Papalampidi et al. (2019) have shown how to identify turning points in feature-length screenplays by projecting synopsis-level annotations.

Crucially, our method does not involve manually annotating turning points in CSI episodes. Instead, we approximate narrative structure automatically by pretraining on the annotations of the TRIPOD dataset of Papalampidi et al. (2019) and employing a variant of their model. We find that narrative structure representations learned on their dataset (which was created for feature-length films), transfer well across cinematic genres and computational tasks. We propose a framework for end-to-end training in which narrative structure is treated as a latent variable for summarization. We extend the CSI dataset (Frermann et al., 2018) with binary labels indicating whether a scene should be included in the summary and present experiments with both supervised and unsupervised summarization models. An overview of our approach is shown in Figure 2.

Our contributions can be summarized as follows: (a) we develop methods for instilling knowledge about narrative structure into generic su-

pervised and unsupervised summarization algorithms; (b) we provide a new layer of annotations for the CSI corpus, which can be used for research in long-form summarization; and (c) we demonstrate that narrative structure can facilitate screenplay summarization; our analysis shows that key events identified in the latent space correlate with important summary content.

2 Related Work

A large body of previous work has focused on the computational analysis of narratives (Mani, 2012; Richards et al., 2009). Attempts to analyze how stories are written have been based on sequences of events (Schank and Abelson, 1975; Chambers and Jurafsky, 2009), plot units (McIntyre and Lapata, 2010; Goyal et al., 2010; Finlayson, 2012) and their structure (Lehnert, 1981; Rumelhart, 1980), as well as on characters or personas in a narrative (Black and Wilensky, 1979; Propp, 1968; Bamman et al., 2014, 2013; Valls-Vargas et al., 2014) and their relationships (Elson et al., 2010; Agarwal et al., 2014; Srivastava et al., 2016).

As mentioned earlier, work on summarization of narratives has had limited appeal, possibly due to the lack of annotated data for modeling and evaluation. Kazantseva and Szpakowicz (2010) summarize short stories based on importance criteria (e.g., whether a segment contains protagonist or location information); they create summaries to help readers decide whether they are interested in reading the whole story, without revealing its plot. Mihalcea and Ceylan (2007) summarize books with an unsupervised graph-based approach operating over segments (i.e., topical units). Their algorithm first generates a summary for each segment and then an overall summary by collecting sentences from the individual segment summaries.

Focusing on screenplays, Gorinski and Lapata (2015) generate a summary by extracting an optimal chain of scenes via a graph-based approach centered around the main characters. In a similar fashion, Tsoneva et al. (2007) create video summaries for TV series episodes; their algorithm ranks sub-scenes in terms of importance using features based on character graphs and textual cues available in the subtitles and movie scripts. Vicol et al. (2018) introduce the MovieGraphs dataset, which also uses character-centered graphs to describe the content of movie video clips.

Our work synthesizes various strands of research on narrative structure analysis (Cutting, 2016; Hauge, 2017), screenplay summarization (Gorinski and Lapata, 2015), and neural network modeling (Dong, 2018). We focus on extractive summarization and our goal is to identify an optimal sequence of key events in a narrative. We aim to create summaries which re-tell the plot of a story in a concise manner. Inspired by recent neural network-based approaches (Cheng and Lapata, 2016; Nallapati et al., 2017; Zhou et al., 2018; Zheng and Lapata, 2019), we develop supervised and unsupervised models for our summarization task based on neural representations of scenes and how these relate to the screenplay's narrative structure. Contrary to most previous work which has focused on characters, we select summary scenes based on events and their importance in the story. Our definition of narrative structure closely follows Papalampidi et al. (2019). However, the model architectures we propose are general and could be adapted to different plot analysis schemes (Field, 2005; Vogler, 2007). To overcome the difficulties in evaluating summaries for longer narratives, we also release a corpus of screenplays with scenes labeled as important (summary worthy). Our annotations augment an existing dataset based on CSI episodes (Frermann et al., 2018), which was originally developed for incremental natural language understanding.

3 Problem Formulation

Let $\mathcal{D}$ denote a screenplay consisting of a sequence of scenes $\mathcal{D} = \{s_1, s_2, \ldots, s_n\}$. Our aim is to select a subset $\mathcal{D}' = \{s_i, \ldots, s_k\}$ consisting of the most *informative* scenes (where $k < n$). Note that this definition produces extractive summaries; we further assume that selected scenes are presented according to their order in the screenplay. We next discuss how summaries can be created using both unsupervised and supervised approaches, and then move on to explain how these are adapted to incorporate narrative structure.

3.1 Unsupervised Screenplay Summarization

Our unsupervised model is based on an extension of TextRank (Mihalcea and Tarau, 2004; Zheng and Lapata, 2019), a well-known algorithm for extractive single-document summarization. In our setting, a screenplay is represented as a graph, in which nodes correspond to scenes and edges between scenes s_i and s_j are weighted by their simi-

larity e_{ij}. A node's centrality (importance) is measured by computing its degree:

$$centrality(s_i) = \lambda_1 \sum_{j<i} e_{ij} + \lambda_2 \sum_{j>i} e_{ij} \quad (1)$$

where $\lambda_1 + \lambda_2 = 1$. The modification introduced in Zheng and Lapata (2019) takes directed edges into account, capturing the intuition that the centrality of any two nodes is influenced by their relative position. Also note that the edges of preceding and following scenes are differentially weighted by λ_1 and λ_2.

Although earlier implementations of TEXTRANK (Mihalcea and Tarau, 2004) compute node similarity based on symbolic representations such as tf*idf, we adopt a neural approach. Specifically, we obtain sentence representations based on a pre-trained encoder. In our experiments, we rely on the Universal Sentence Encoder (USE; Cer et al. 2018), however, other embeddings are possible.[1] We represent a scene by the mean of its sentence representations and measure scene similarity e_{ij} using cosine.[2] As in the original TEXTRANK algorithm (Mihalcea and Tarau, 2004), scenes are ranked based on their centrality and the M most central ones are selected to appear in the summary.

3.2 Supervised Screenplay Summarization

Most extractive models frame summarization as a classification problem. Following a recent approach (SUMMARUNNER; Nallapati et al. 2017), we use a neural network-based encoder to build representations for scenes and apply a binary classifier over these to predict whether they should be in the summary. For each scene $s_i \in \mathcal{D}$, we predict a label $y_i \in \{0,1\}$ (where 1 means that s_i must be in the summary) and assign a score $p(y_i|s_i,\mathcal{D},\theta)$ quantifying s_i's relevance to the summary (θ denotes model parameters). We assemble a summary by selecting M sentences with the top $p(1|s_i,\mathcal{D},\theta)$.

We calculate sentence representations via the pre-trained USE encoder (Cer et al., 2018); a scene is represented as the weighted sum of the representations of its sentences, which we obtain from a BiLSTM equipped with an attention mechanism. Next, we compute richer scene representations by modeling surrounding context of a given scene. We encode the screenplay with a BiLSTM network and obtain contextualized representations s'_i for scenes s_i by concatenating the hidden layers of the forward $\overrightarrow{h_i}$ and backward $\overleftarrow{h_i}$ LSTM, respectively: $s'_i = [\overrightarrow{h_i}; \overleftarrow{h_i}]$. The vector s'_i therefore represents the *content* of the i^{th} scene.

We also estimate the *salience* of scene s_i by measuring its similarity with a global screenplay content representation d. The latter is the weighted sum of all scene representations $s_1, s_2, \ldots, s_n$. We calculate the semantic similarity between s'_i and d by computing the element-wise dot product b_i, cosine similarity c_i, and pairwise distance u_i between their respective vectors:

$$b_i = s'_i \odot d \quad c_i = \frac{s'_i \cdot d}{\|s'_i\| \|d\|} \quad (2)$$

$$u_i = \frac{s'_i \cdot d}{\max(\|s'_i\|_2 \cdot \|d\|_2)} \quad (3)$$

The salience v_i of scene s_i is the concatenation of the similarity metrics: $v_i = [b_i; c_i; u_i]$. The content vector s'_i and the salience vector v_i are concatenated and fed to a single neuron that outputs the probability of a scene belonging to the summary.[3]

3.3 Narrative Structure

We now explain how to inject knowledge about narrative structure into our summarization models. For both models, such knowledge is transferred via a network pre-trained on the TRIPOD[4] dataset introduced by Papalampidi et al. (2019). This dataset contains 99 movies annotated with turning points. TPs are key events in a narrative that define the progression of the plot and occur between consecutive acts (thematic units). It is often assumed (Cutting, 2016) that there are six acts in a film (Figure 1), each delineated by a turning point (arrows in the figure). Each of the five TPs has also a well-defined function in the narrative: we present each TP alongside with its definition as stated in screenwriting theory (Hauge, 2017) and adopted by Papalampidi et al. (2019) in Table 1 (see Appendix A for a more detailed description of narrative structure theory). Papalampidi et al. (2019) identify scenes in movies that correspond to these key events as a means for analyzing the narrative

[1] USE performed better than BERT in our experiments.

[2] We found cosine to be particularly effective with USE representations; other metrics are also possible.

[3] Aside from salience and content, Nallapati et al. (2017) take into account novelty and position-related features. We ignore these as they are specific to news articles and denote the modified model as SUMMARUNNER*.

[4] https://github.com/ppapalampidi/TRIPOD

Turning Point	Definition
TP1: Opportunity	Introductory event that occurs after the presentation of the story setting.
TP2: Change of Plans	Event where the main goal of the story is defined.
TP3: Point of No Return	Event that pushes the main character(s) to fully commit to their goal.
TP4: Major Setback	Event where everything falls apart (temporarily or permanently).
TP5: Climax	Final event of the main story, moment of resolution.

Table 1: Turning points and their definitions as given in Papalampidi et al. (2019)

structure of movies. They collect sentence-level TP annotations for plot synopses and subsequently project them via distant supervision onto screenplays, thereby creating silver-standard labels. We utilize this silver-standard dataset in order to pretrain a network which performs TP identification.

TP Identification Network We first encode screenplay scenes via a BiLSTM equipped with an attention mechanism. We then contextualize them with respect to the whole screenplay via a second BiLSTM. Next, we compute topic-aware scene representations t_i via a context interaction layer (CIL) as proposed in Papalampidi et al. (2019). CIL is inspired by traditional segmentation approaches (Hearst, 1997) and measures the semantic similarity of the current scene with a preceding and following context window in the screenplay. Hence, the topic-aware scene representations also encode the degree to which each scene acts as a topic boundary in the screenplay.

In the final layer, we employ TP-specific attention mechanisms to compute the probability p_{ij} that scene t_i represents the j^{th} TP in the screenplay. Note that we expect the TP-specific attention distributions to be sparse, as there are only a few scenes which are relevant for a TP (recall that TPs are boundary scenes between sections). To encourage sparsity, we add a low temperature value τ (Hinton et al., 2015) to the softmax part of the attention mechanisms:

$$g_{ij} = \tanh(W_j t_i + b_j), \quad g_j \in [-1,1] \tag{4}$$

$$p_{ij} = \frac{\exp(g_{ij}/\tau)}{\sum_{t=1}^{T}\exp(g_{tj}/\tau)}, \quad \sum_{i=1}^{T} p_{ij} = 1 \tag{5}$$

where W_j, b_j represent the trainable weights of the attention layer of the j^{th} TP.

Unsupervised SUMMER We now introduce our model, SUMMER (short for Screenplay Summarization with Narrative Structure).[5] We first present an unsupervised variant which modifies the computation of scene centrality in the directed version of TEXTRANK (Equation (1)).

Specifically, we use the pre-trained network described in Section 3.3 to obtain TP-specific attention distributions. We then select an overall score f_i for each scene (denoting how likely it is to act as a TP). We set $f_i = \max_{j\in[1,5]} p_{ij}$, i.e., to the p_{ij} value that is highest across TPs. We incorporate these scores into centrality as follows:

$$centrality(s_i) = \lambda_1 \sum_{j<i}(e_{ij}+f_j) + \lambda_2 \sum_{j>i}(e_{ij}+f_i) \tag{6}$$

Intuitively, we add the f_j term in the forward sum in order to incrementally increase the centrality scores of scenes as the story moves on and we encounter more TP events (i.e., we move to later sections in the narrative). At the same time, we add the f_i term in the backward sum in order to also increase the scores of scenes identified as TPs.

Supervised SUMMER We also propose a supervised variant of SUMMER following the basic model formulation in Section 3.3. We still represent a scene as the concatenation of a content vector s' and salience vector v', which serve as input to a binary classifier. However, we now modify how salience is determined; instead of computing a general global content representation d for the screenplay, we identify a sequence of TPs and measure the semantic similarity of each scene with this sequence. Our model is depicted in Figure 3.

We utilize the pre-trained TP network (Figures 3(a) and (b)) to compute sparse attention scores over scenes. In the supervised setting, where gold-standard binary labels provide a training signal, we fine-tune the network in an end-to-end fashion on summarization (Figure 3(c)). We compute the TP representations via the attention scores; we calculate a vector tp_j as the weighted sum of all topic-aware scene representations t produced via CIL: $tp_j = \sum_{i\in[1,N]} p_{ij} t_i$, where N is the number of scenes in a screenplay. In practice, only a few scenes contribute to tp_j due to the τ parameter in the softmax function (Equation (5)).

A TP-scene interaction layer measures the semantic similarity between scenes t_i and latent TP representations tp_j (Figure 3(c)). Intuitively, a complete summary should contain scenes which

[5] We make our code publicly available at https://github.com/ppapalampidi/SUMMER.

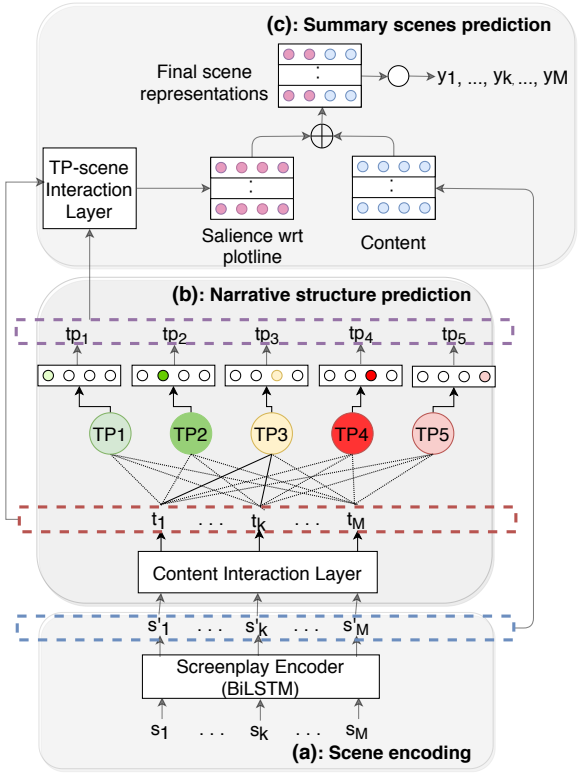

Figure 3: Overview of SUMMER. We use one TP-specific attention mechanism per turning point in order to acquire TP-specific distributions over scenes. We then compute the similarity between TPs and contextualized scene representations. Finally, we perform max pooling over TP-specific similarity vectors and concatenate the final similarity representation with the contextualized scene representation.

are related to at least one of the key events in the screenplay. We calculate the semantic similarity v_{ij} of scene t_i with TP tp_j as in Equations (2) and (3). We then perform max pooling over vectors $v_{i1}, \ldots, v_{iT}$, where T is the number of TPs (i.e., five) and calculate a final similarity vector v'_i for the i^{th} scene.

The model is trained end-to-end on the summarization task using *BCE*, the binary cross-entropy loss function. We add an extra regularization term to this objective to encourage the TP-specific attention distributions to be orthogonal (since we want each attention layer to attend to different parts of the screenplay). We thus maximize the Kullback-Leibler (KL) divergence $\mathcal{D}_{KL}$ between all pairs of TP attention distributions tp_i, $i \in [1,5]$:

$$O = \sum_{i\in[1,5]} \sum_{j\in[1,5], j\neq i} \log \frac{1}{\mathcal{D}_{KL}\left(tp_i \| tp_j\right) + \varepsilon} \quad (7)$$

Furthermore, we know from screenwriting theory (Hauge, 2017) that there are rules of thumb as to when a TP should occur (e.g., the Opportunity occurs after the first 10% of a screenplay, Change of Plans is approximately 25% in). It is reasonable to discourage tp distributions to deviate drastically from these expected positions. Focal regularization F minimizes the KL divergence $\mathcal{D}_{KL}$ between each TP attention distribution tp_i and its expected position distribution th_i:

$$F = \sum_{i\in[1,5]} \mathcal{D}_{KL}\left(tp_i \| th_i\right) \quad (8)$$

The final loss $\mathcal{L}$ is the weighted sum of all three components, where a, b are fixed during training: $\mathcal{L} = BCE + aO + bF$.

4 Experimental Setup

Crime Scene Investigation Dataset We performed experiments on an extension of the CSI dataset[6] introduced by Frermann et al. (2018). It consists of 39 CSI episodes, each annotated with *word-level* labels denoting whether the perpetrator is mentioned in the utterances characters speak. We further collected *scene-level* binary labels indicating whether episode scenes are important and should be included in a summary. Three human judges performed the annotation task after watching the CSI episodes scene-by-scene. To facilitate the annotation, judges were asked to indicate why they thought a scene was important, citing the following reasons: it revealed (i) the victim, (ii) the cause of death, (iii) an autopsy report, (iv) crucial evidence, (v) the perpetrator, and (vi) the motive or the relation between perpetrator and victim. Annotators were free to select more than one or none of the listed reasons where appropriate. We can think of these reasons as high-level *aspects* a good summary should cover (for CSI and related crime series). Annotators were not given any information about TPs or narrative structure; the annotation was not guided by theoretical considerations, rather our aim was to produce useful CSI summaries. Table 2 presents the dataset statistics (see also Appendix B for more detail).

Implementation Details In order to set the hyperparameters of all proposed networks, we used a small development set of four episodes from the CSI dataset (see Appendix B for details). After experimentation, we set the temperature τ of the softmax layers for the TP-specific attentions (Equation (5)) to 0.01. Since the binary labels in the

[6] https://github.com/EdinburghNLP/csi-corpus

overall	
episodes	39
scenes	1544
summary scenes	454
per episode	
scenes	39.58 (6.52)
crime-specific aspects	5.62 (0.24)
summary scenes	11.64 (2.98)
summary scenes (%)	29.75 (7.35)
sentences	822.56 (936.23)
tokens	13.27k (14.67k)
per episode scene	
sentences	20.78 (35.61)
tokens	335.19 (547.61)
tokens per sentence	16.13 (16.32)

Table 2: CSI dataset statistics; means and (std).

supervised setting are imbalanced, we apply class weights to the binary cross-entropy loss of the respective models. We weight each class by its inverse frequency in the training set. Finally, in supervised SUMMER, where we also identify the narrative structure of the screenplays, we consider as key events per TP the scenes that correspond to an attention score higher than 0.05. More implementation details can be found in Appendix C.

As shown in Table 2, the gold-standard summaries in our dataset have a compression rate of approximately 30%. During inference, we select the top M scenes as the summary, such that they correspond to 30% of the length of the episode.

5 Results and Analysis

Is Narrative Structure Helpful? We perform 10-fold cross-validation and evaluate model performance in terms of F1 score. Table 3 summarizes the results of unsupervised models. We present the following baselines: Lead 30% selects the first 30% of an episode as the summary, Last 30% selects the last 30%, and Mixed 30%, randomly selects 15% of the summary from the first 30% of an episode and 15% from the last 30%. We also compare SUMMER against TEXTRANK based on tf*idf (Mihalcea and Tarau, 2004), the directed neural variant described in Section 3.1 without any TP information, a variant where TPs are approximated by their expected position as postulated in screenwriting theory, and a variant that incorporates information about characters (Gorinski and Lapata, 2015) instead of narrative structure. For the character-based TEXTRANK, called SCENESUM, we substitute the f_i, f_j scores in Equation (6) with character-related importance scores c_i similar to the definition in Gorinski and Lapata (2015):

$$c_i = \frac{\sum_{c \in C} [c \in S \cup main(C)]}{\sum_{c \in C} [c \in S]} \quad (9)$$

where S is the set of all characters participating in scene s_i, C is the set of all characters participating in the screenplay and $main(C)$ are all the main characters of the screenplay. We retrieve the set of main characters from the IMDb page of the respective episode. We also note that human agreement for our task is 79.26 F1 score, as measured on a small subset of the corpus.

Model	F1
Lead 30%	30.66
Last 30%	39.85
Mixed 30%	34.32
TEXTRANK, undirected, tf*idf	32.11
TEXTRANK, directed, neural	41.75
TEXTRANK, directed, expected TP positions	41.05
SCENESUM, directed, character-based weights	42.02
SUMMER	**44.70**

Table 3: Unsupervised screenplay summarization.

	F1	Coverage of aspects	# scenes per TP
Lead 30%	30.66	–	–
Last 30%	39.85	–	–
Mixed 30%	34.32	–	–
SUMMARUNNER*	48.56	–	–
SCENESUM	47.71	–	–
SUMMER, fixed one-hot TPs	46.92	63.11	1.00
SUMMER, fixed distributions	47.64	67.01	1.05
SUMMER, −P, −R	**51.93**	44.48	1.19
SUMMER, −P, +R	49.98	51.96	1.14
SUMMER, +P, −R	50.56	62.35	3.07
SUMMER, +P, +R	**52.00**	**70.25**	1.20

Table 4: Supervised screenplay summarization; for in SUMMER variants, we also report the percentage of aspect labels covered by latent TP predictions.

As shown in Table 3, SUMMER achieves the best performance (44.70 F1 score) among all models and is superior to an equivalent model which uses expected TP positions or a character-based representation. This indicates that the pre-trained network provides better predictions for key events than position and character heuristics, even though there is a domain shift from Hollywood movies in the TRIPOD corpus to episodes of a crime series in the CSI corpus. Moreover, we find that the directed versions of TEXTRANK are better at identifying important scenes than the undirected version. We found that performance peaks with $\lambda_1 = 0.7$ (see Equation (6)), indicating that higher importance is given to scenes as the story progresses (see Appendix D for experiments with different λ values).

In Table 4, we report results for supervised models. Aside from the various baselines in the first block of the table, we compare the neural extractive model SUMMARUNNER*[7] (Nallapati et al., 2017) presented in Section 3.2 with several variants of our model SUMMER. We experimented with randomly initializing the network for TP identification (−P) and with using a pre-trained network (+P). We also experimented with removing the regularization terms, O and F (Equations (7) and (8)) from the loss (−R). We assess the performance of SUMMER when we follow a two-step approach where we first predict TPs via the pre-trained network and then train a network on screenplay summarization based on fixed TP representations (fixed one-hot TPs), or alternatively use expected TP position distributions as postulated in screenwriting theory (fixed distributions). Finally, we incorporate character-based information into our baseline and create a supervised version of SCENESUM. We now utilize the character importance scores per scene (Equation (9)) as attention scores – instead of using a trainable attention mechanism – when computing the global screenplay representation d (Section 3.2).

Table 4 shows that all end-to-end SUMMER variants outperform SUMMARUNNER*. The best result (52.00 F1 Score) is achieved by pre-trained SUMMER with regularization, outperforming SUMMARUNNER* by an absolute difference of 3.44. The randomly initialized version with no regularization achieves similar performance (51.93 F1 score). For summarizing screenplays, explicitly encoding narrative structure seems to be more beneficial than general representations of scene importance. Finally, two-step versions of SUMMER perform poorly, which indicates that end-to-end training and fine-tuning of the TP identification network on the target dataset is crucial.

What Does the Model Learn? Apart from performance on summarization, we would also like to examine the quality of the TPs inferred by SUMMER (supervised variant). Problematically, we do not have any gold-standard TP annotation in the CSI corpus. Nevertheless, we can implicitly assess whether they are meaningful by measuring how well they correlate with the reasons annotators cite to justify their decision to include a scene in the summary (e.g., because it reveals cause of death or provides important evidence). Specifically, we compute the extent to which these aspects overlap with the TPs predicted by SUMMER as:

$$C = \frac{\sum_{A_i \in A} \sum_{TP_j \in TP} [dist(TP_j, A_i) \leq 1]}{|A|} \quad (10)$$

where A is the set of all aspect scenes, $|A|$ is the number of aspects, TP is the set of scenes inferred as TPs by the model, A_i and TP_j are the subsets of scenes corresponding to the i^{th} aspect and j^{th} TP, respectively, and $dist(TP_j, A_i)$ is the minimum distance between TP_j and A_i in number of scenes.

The proportion of aspects covered is given in Table 4, middle column. We find that coverage is relatively low (44.48%) for the randomly initialized SUMMER with no regularization. There is a slight improvement of 7.48% when we force the TP-specific attention distributions to be orthogonal and close to expected positions. Pre-training and regularization provide a significant boost, increasing coverage to 70.25%, while pre-trained SUMMER without regularization infers on average more scenes representative of each TP. This shows that the orthogonal constraint also encourages sparse attention distributions for TPs.

Table 5 shows the degree of association between individual TPs and summary aspects (see Appendix D for illustrated examples). We observe that Opportunity and Change of Plans are mostly associated with information about the crime scene and the victim, Climax is focused on the revelation of the motive, while information relating to cause of death, perpetrator, and evidence is captured by both Point of no Return and Major Setback. Overall, the generic Hollywood-inspired TP labels are adjusted to our genre and describe crime-related key events, even though no aspect labels were provided to our model during training.

Do Humans Like the Summaries? We also conducted a human evaluation experiment using the summaries created for 10 CSI episodes.[8] We produced summaries based on the gold-standard annotations (Gold), SUMMARUNNER*, and the supervised version of SUMMER. Since 30% of an episode results in lengthy summaries (15 minutes on average), we further increased the compression rate for this experiment by limiting each summary to six scenes. For the gold standard condition, we randomly selected exactly one scene

[7] Our adaptation of SUMMARUNNER that considers content and salience vectors for scene selection.

[8] https://github.com/ppapalampidi/SUMMER/tree/master/video_summaries

Turning Point	Crime scene	Victim	Death Cause	Perpetrator	Evidence	Motive
Opportunity	**56.76**	**52.63**	15.63	15.38	2.56	0.00
Change of Plans	**27.03**	**42.11**	**21.88**	15.38	5.13	0.00
Point of no Return	8.11	13.16	9.38	**25.64**	**48.72**	5.88
Major Setback	0.00	0.00	6.25	10.25	**48.72**	**35.29**
Climax	2.70	0.00	6.25	2.56	**23.08**	**55.88**

Table 5: Percentage of aspect labels covered per TP for SUMMER, +P, +R.

System	Crime scene	Victim	Death Cause	Perpetrator	Evidence	Motive	Overall	Rank
SUMMARUNNER*	85.71	**93.88**	75.51	81.63	59.18	38.78	72.45	2.18
SUMMER	**89.80**	87.76	**83.67**	81.63	**77.55**	**57.14**	**79.59**	2.00
Gold	**89.80**	91.84	71.43	**83.67**	65.31	**57.14**	76.53	1.82

Table 6: Human evaluation: percentage of yes answers by AMT workers regarding each aspect in a summary. All differences in (average) Rank are significant ($p < 0.05$, using a χ^2 test).

per aspect. For SUMMARUNNER* and SUMMER we selected the top six predicted scenes based on their posterior probabilities. We then created video summaries by isolating and merging the selected scenes in the raw video.

We asked Amazon Mechanical Turk (AMT) workers to watch the video summaries for all systems and rank them from most to least informative. They were also presented with six questions relating to the aspects the summary was supposed to cover (e.g., Was the victim revealed in the summary? Do you know who the perpetrator was?). They could answer Yes, No, or Unsure. Five workers evaluated each summary.

Table 6 shows the proportion of times participants responded Yes for each aspect across the three systems. Although SUMMER does not improve over SUMMARUNNER* in identifying basic information (i.e., about the victim and perpetrator), it creates better summaries overall with more diverse content (i.e., it more frequently includes information about cause of death, evidence, and motive). This observation validates our assumption that identifying scenes that are semantically close to the key events of a screenplay leads to more complete and detailed summaries. Finally, Table 6 also lists the average rank per system (lower is better), which shows that crowdworkers like gold summaries best, SUMMER is often ranked second, followed by SUMMARUNNER* in third place.

6 Conclusions

In this paper we argued that the underlying structure of narratives is beneficial for long-form summarization. We adapted a scheme for identifying narrative structure (i.e., turning points) in Hollywood movies and showed how this information can be integrated with supervised and unsupervised extractive summarization algorithms. Experiments on the CSI corpus showed that this scheme transfers well to a different genre (crime investigation) and that utilizing narrative structure boosts summarization performance, leading to more complete and diverse summaries. Analysis of model output further revealed that latent events encapsulated by turning points correlate with important aspects of a CSI summary.

Although currently our approach relies solely on textual information, it would be interesting to incorporate additional modalities such as video or audio. Audiovisual information could facilitate the identification of key events and scenes. Besides narrative structure, we would also like to examine the role of *emotional arcs* (Vonnegut, 1981; Reagan et al., 2016) in a screenplay. An often integral part of a compelling story is the emotional experience that is evoked in the reader or viewer (e.g., somebody gets into trouble and then out of it, somebody finds something wonderful, loses it, and then finds it again). Understanding emotional arcs may be useful to revealing a story's shape, highlighting important scenes, and tracking how the story develops for different characters over time.

Acknowledgments

We thank the anonymous reviewers for their feedback. We gratefully acknowledge the support of the European Research Council (Lapata; award 681760, "Translating Multiple Modalities into Text") and of the Leverhulme Trust (Keller; award IAF-2017-019).

References

Apoorv Agarwal, Sriramkumar Balasubramanian, Jiehan Zheng, and Sarthak Dash. 2014. Parsing

Screenplays for Extracting Social Networks from Movies. In *Proceedings of the 3rd Workshop on Computational Linguistics for Literature*, pages 50–58, Gothenburg, Sweden.

David Bamman, Brendan O'Connor, and Noah A. Smith. 2013. Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 352–361, Sofia, Bulgaria.

David Bamman, Ted Underwood, and Noah A. Smith. 2014. A Bayesian mixed effects model of literary character. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 370–379, Baltimore, Maryland.

John B Black and Robert Wilensky. 1979. An evaluation of story grammars. *Cognitive science*, 3(3):213–229.

Charles Oscar Brink. 2011. *Horace on Poetry: The'Ars Poetica'*. Cambridge University Press.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 602–610, Suntec, Singapore.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494.

James E Cutting. 2016. Narrative theory and the dynamics of popular movies. *Psychonomic bulletin & review*, 23(6):1713–1743.

Yue Dong. 2018. A survey on neural network-based summarization methods. *ArXiv*, abs/1804.04589.

David K. Elson, Nicholas Dames, and Kathleen R. Mckeown. 2010. Extracting social networks from literary fiction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 138–147, Uppsala, Sweden.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Syd Field. 2005. *Screenplay: Foundations of Screenwriting*. Dell Publishing Company.

Mark Alan Finlayson. 2012. *Learning Narrative Structure from Annotated Folktales*. Ph.D. thesis, Massachusetts Institute of Technology.

Lea Frermann, Shay B Cohen, and Mirella Lapata. 2018. Whodunnit? crime drama as a case for natural language understanding. *Transactions of the Association of Computational Linguistics*, 6:1–15.

Gustav Freytag. 1896. *Freytag's technique of the drama: an exposition of dramatic composition and art*. Scholarly Press.

Philip John Gorinski and Mirella Lapata. 2015. Movie script summarization as graph-based scene extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1066–1076, Denver, Colorado. Association for Computational Linguistics.

Amit Goyal, Ellen Riloff, and Hal Daumé III. 2010. Automatically producing plot unit representations for narrative text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 77–86, Cambridge, MA.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. NEWSROOM: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 708–719, New Orleans, USA.

Michael Hauge. 2017. *Storytelling Made Easy: Persuade and Transform Your Audiences, Buyers, and Clients – Simply, Quickly, and Profitably*. Indie Books International.

Marti A Hearst. 1997. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational linguistics*, 23(1):33–64.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28*, pages 1693–1701. Morgan, Kaufmann.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Anna Kazantseva and Stan Szpakowicz. 2010. Summarizing short stories. *Computational Linguistics*, 36(1):71–109.

Joy Kim and Andrés Monroy-Hernández. 2015. Storia: Summarizing social media content based on narrative theory using crowdsourcing. *CoRR*, abs/1509.03026.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Wendy G. Lehnert. 1981. Plot units and narrative summarization. *Cognitive Science*, 5(4):293–331.

Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, Florence, Italy. Association for Computational Linguistics.

Interjeet Mani. 2012. *Computational Modeling of Narative*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool Publishers.

Neil McIntyre and Mirella Lapata. 2010. Plot induction and evolutionary search for story generation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1562–1572, Uppsala, Sweden.

Rada Mihalcea and Hakan Ceylan. 2007. Explorations in automatic book summarization. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 380–389.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Pinelopi Papalampidi, Frank Keller, and Mirella Lapata. 2019. Movie plot analysis via turning point identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1707–1717.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Patrice Pavis. 1998. *Dictionary of the theatre: Terms, concepts, and analysis*. University of Toronto Press.

Vladimir Iakovlevich Propp. 1968. *Morphology of the Folktale*. University of Texas.

Andrew J. Reagan, Lewis Mitchell, Dilan Kiley, Christopher M. Danforth, and Peter Sheridan Dodds. 2016. The emotional arcs of stories are dominated by six basic shapes. *EPJ Data Science*, 5(31):1–12.

Whitman Richards, Mark Alan Finlayson, and Patrick Henry Winston. 2009. Advancing computational models of narrative. Technical Report 63:2009, MIT Computer Science and Atrificial Intelligence Laboratory.

David E. Rumelhart. 1980. On evaluating story grammars. *Cognitive Science*, 4(3):313–316.

Evan Sandhaus. 2008. The New York Times Annotated Corpus. *Linguistic Data Consortium, Philadelphia*, 6(12).

Roger C. Schank and Robert P. Abelson. 1975. Scripts, plans, and knowledge. In *Proceedings of the 4th International Joint Conference on Artificial Intelligence*, pages 151–157, Tblisi, USSR.

Shashank Srivastava, Snigdha Chaturvedi, and Tom Mitchell. 2016. Inferring interpersonal relations in narrative summaries. In *Proceedings of the 13th AAAI Conference on Artificial Intelligence*, pages 2807–2813, Phoenix, Arizona. AAAI Press.

Kristin Thompson. 1999. *Storytelling in the new Hollywood: Understanding classical narrative technique*. Harvard University Press.

Tsvetomira Tsoneva, Mauro Barbieri, and Hans Weda. 2007. Automated summarization of narrative video on a semantic level. In *International Conference on Semantic Computing (ICSC 2007)*, pages 169–176. IEEE.

Josep Valls-Vargas, J. Zhu, and Santiago Ontanon. 2014. Toward automatic role identification in unannotated folk tales. In *Proceedings of the 10th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, pages 188–194.

Paul Vicol, Makarand Tapaswi, Lluis Castrejon, and Sanja Fidler. 2018. Moviegraphs: Towards understanding human-centric situations from videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8581–8590.

Christopher Vogler. 2007. *Writer's Journey: Mythic Structure for Writers*. Michael Wiese Productions.

Kurt Vonnegut. 1981. *Palm Sunday*. RosettaBooks LLC, New York.

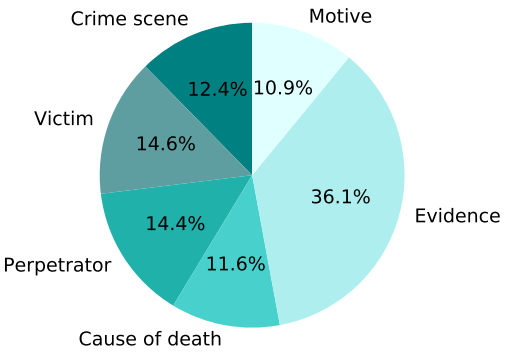

Figure 4: Average composition of a CSI summary based on different crime-related aspects.

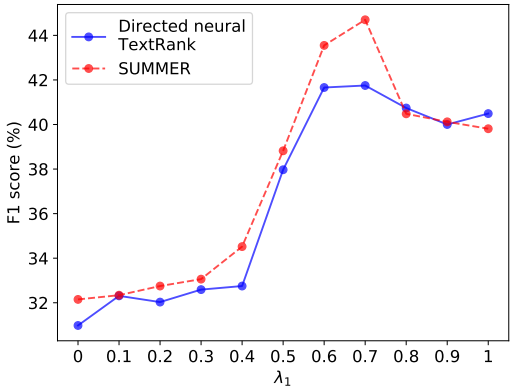

Figure 5: F1 score (%) for directed neural TEXTRANK and SUMMER for unsupervised summarization with respect to different λ_1 values. Higher λ_1 values correspond to higher importance in the next context for the centrality computation of a current scene.

Hao Zheng and Mirella Lapata. 2019. Sentence centrality revisited for unsupervised summarization. *arXiv preprint arXiv:1906.03508*.

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. Neural document summarization by jointly learning to score and select sentences. *arXiv preprint arXiv:1807.02305*.

A Narrative Structure Theory

The initial formulation of narrative structure was promoted by Aristotle, who defined the basic triangle-shaped plot structure, that has a beginning (*protasis*), middle (*epitasis*) and end (*catastrophe*) (Pavis, 1998). However, later theories argued that the structure of a play should be more complex (Brink, 2011) and hence, other schemes (Freytag, 1896) were proposed with fine-grained stages and events defining the progression of the plot. These events are considered as the precursor of turning points, defined by Thompson (1999) and used in modern variations of screenplay theory. Turning points are narrative moments from which the plot goes in a different direction. By definition these occur at the junctions of acts.

Currently, there are myriad schemes describing the narrative structure of films, which are often used as a practical guide for screenwriters (Cutting, 2016). One variation of these modern schemes is adopted by Papalampidi et al. (2019), who focus on the definition of turning points and demonstrate that such events indeed exist in films and can be automatically identified. According to the adopted scheme (Hauge, 2017), there are six stages (acts) in a film, namely *the setup*, *the new situation*, *progress*, *complications and higher stakes*, *the final push* and *the aftermath*, separated by the five turning points presented in Table 1.

B CSI Corpus

As described in Section 4, we collected aspect-based summary labels for all episodes in the CSI corpus. In Figure 4 we illustrate the average composition of a summary based on the different aspects seen in a crime investigation (e.g., crime scene, victim, cause of death, perpetrator, evidence). Most of these aspects are covered in 10–15% of a summary, which corresponds to approximately two scenes in the episode. Only the "Evidence" aspect occupies a larger proportion of the summary (36.1%) corresponding to five scenes. However, there exist scenes which cover multiple aspects (an as a result are annotated with more than one label) and episodes that do not include any scenes related to a specific aspect (e.g., if the murder was a suicide, there is no perpetrator).

We should note that Frermann et al. (2018) discriminate between different cases presented in the same episode in the original CSI dataset. Specifically, there are episodes in the dataset, where except for the primary crime investigation case, a second one is presented occupying a significantly smaller part of the episode. Although in the original dataset, there are annotations available indicating which scenes refer to each case, we assume no such knowledge treating the screenplay as a single unit — most TV series and movies contain sub-stories. We also hypothesize that the latent identified TP events in SUMMER should relate to the primary case.

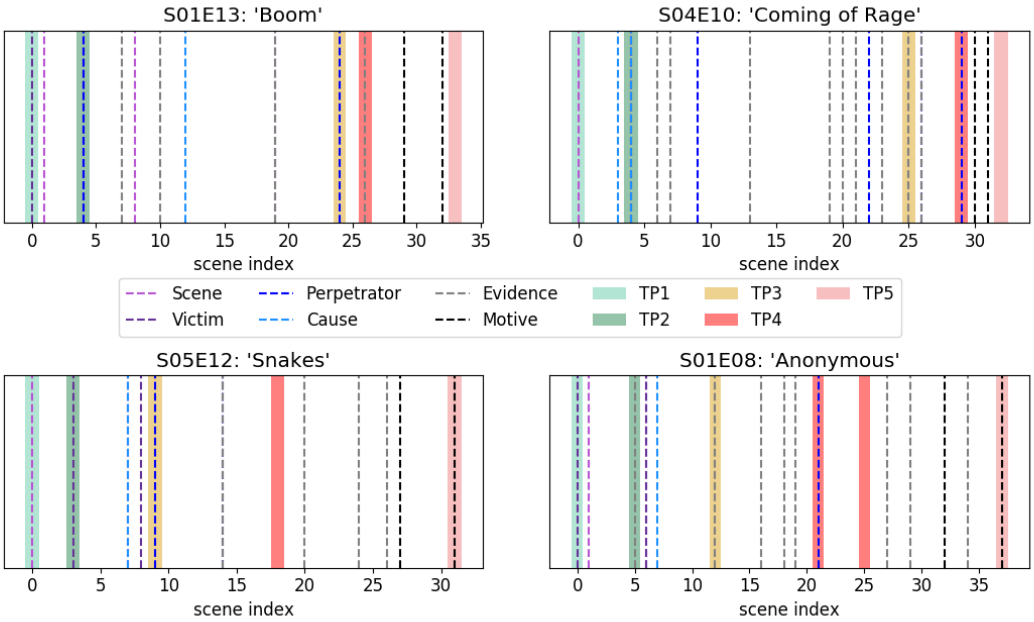

Figure 6: Examples of inferred TPs alongside with gold-standard aspect-based summary labels in CSI episodes at test time. The TP events are identified in the latent space for the supervised version of SUMMER (+P, +R).

C Implementation Details

In all unsupervised versions of TEXTRANK and SUMMER we used a threshold h equal to 0.2 for removing weak edges from the corresponding fully connected screenplay graphs. For the supervised version of SUMMER, where we use additional regularization terms in the loss function, we experimentally set the weights a and b for the different terms to 0.15 and 0.1, respectively.

We used the Adam algorithm (Kingma and Ba, 2014) for optimizing our networks. After experimentation, we chose an LSTM with 64 neurons for encoding the scenes in the screenplay and another identical one for contextualizing them. For the context interaction layer, the window l for computing the surrounding context of a screenplay scene was set to 20% of the screenplay length as proposed in Papalampidi et al. (2019). Finally, we also added a dropout of 0.2. For developing our models we used PyTorch (Paszke et al., 2017).

D Additional Results

We illustrate in Figure 5 the performance (F1 score) of the directed neural TEXTRANK and SUMMER models in the unsupervised setting with respect to different λ_1 values. Higher λ_1 values correspond to higher importance for the succeeding scenes and respectively lower importance for the preceding ones, since λ_1 and λ_2 are bounded ($\lambda_1 + \lambda_2 = 1$).

We observe that performance increases when higher importance is attributed to screenplay scenes as the story moves on ($\lambda_1 > 0.5$), whereas for extreme cases ($\lambda_1 \rightarrow 1$), where only the later parts of the story are considered, performance drops. Overall, the same peak appears for both TEXTRANK and SUMMER when $\lambda_1 \in [0.6, 0.7]$, which means that slightly higher importance is attributed to the screenplay scenes that follow. Intuitively, initial scenes of an episode tend to have high similarity with all other scenes in the screenplay, and on their own are not very informative (e.g., the crime, victim, and suspects are introduced but the perpetrator is not yet known). As a result, the undirected version of TEXTRANK tends to favor the first part of the story and the resulting summary consists mainly of initial scenes. By adding extra importance to later scenes, we also encourage the selection of later events that might be surprising (and hence have lower similarity with other scenes) but more informative for the summary. Moreover, in SUMMER, where the weights change in a systematic manner based on narrative structure, we also observe that scenes appearing later in the screenplay are selected more often for inclusion in the summary.

As described in detail in Section 3.3, we also

infer the narrative structure of CSI episodes in the supervised version of SUMMER via latent TP representations. During experimentation (see Section 5), we found that these TPs are highly correlated with different aspects of a CSI summary. In Figure 6 we visualize examples of identified TPs on CSI episodes during test time alongside with gold-standard aspect-based summary annotations. Based on the examples, we empirically observe that different TPs tend to capture different types of information helpful for summarizing crime investigation stories (e.g., crime scene, victim, perpetrator, motive).